

Lista de exercícios - Matrizes

ATENÇÃO!!!! É PROIBIDA a declaração estática das matrizes com VARIÁVEIS nas DIMENSÕES. Apenas utilize constantes ou constantes nomeadas.

```

1
2  int mat[4][5]; OK!!!
3
4  #define L 4
5  #define C 5
6  int mat[L][C];    OK!!!
7
8  int totLin,totCol, mat[totLin][totCol];    PROIBIDO!!!!

```

1. Faça um programa que preencha com valores aleatórios uma matriz de dimensão 5×5 e imprima matriz e depois sua transposta.
2. Faça um programa para gerar um triângulo de Pascal de N linhas em uma matriz de tamanho $N \times N$.

```

      1
    1  1
  1  2  1
1  3  3  1
1  4  6  4  1
1  5 10 10 5  1
1  6 15 20 15 6  1

```

Figura 1: Exemplo de Triângulo de Pascal

3. Na teoria de sistemas define-se elemento minimax de uma matriz, o menor elemento da linha em que se encontra o maior elemento da matriz. Faça um programa que leia uma matriz A de dimensão 10×10 e determine o elemento minimax desta matriz, imprima a linha e coluna que o elemento se encontra, o elemento e o maior elemento da matriz.
4. Dizemos que uma matriz quadrada inteira é um quadrado mágico se a soma dos elementos de cada linha, a soma dos elementos de cada coluna e a soma dos elementos das diagonais principal e secundária são todos iguais. Exemplo:

```

8 0 7
4 5 6
3 10 2

```

é um quadrado mágico pois $8 + 0 + 7 = 4 + 5 + 6 = 3 + 10 + 2 = 8 + 4 + 3 = 0 + 5 + 10 = 7 + 6 + 2 = 8 + 5 + 2 = 3 + 5 + 7$
 Faça um programa, dada uma matriz quadrada A (lida do teclado), imprima na tela se ela é um quadrado mágico ou não. Dica: observe que, assim que uma das somas for diferente, você já pode sair do laço!

5. Dizemos que uma matriz inteira A ($n \times n$) é uma matriz de permutação se em cada linha e em cada coluna houver $n-1$ elementos nulos e um único elemento igual a 1. Dada uma matriz inteira A ($n \times n$) verificar se A é de permutação. Dica: você deve pensar em uma estratégia para contar o número de valores nulos e não nulos!

Matrizes e Funções

6. Faça uma função para gerar automaticamente numeros entre 0 e 99 de uma cartela de bingo. Sabendo que cada cartela devera conter 5 linhas de 5 números cada uma, gere estes dados de modo a nao ter números repetidos dentro das cartelas. Faça a função `main(...)` que chama essa função para criar a cartela e depois chame a função `imprimeMatriz(...)` para imprimir na tela a cartela criada (lembre-se de adaptar a função).

7. Faça um programa que forneça ao usuário um menu de opções (investigue o comando `switch(...)`) para:

- (0) criar duas matrizes 10 x 10 com valores inteiros
- (1) imprimir as matrizes originais
- (2) somar as duas matrizes e imprimir o resultado
- (3) subtrair a primeira matriz da segunda e imprimir o resultado
- (4) multiplicar uma constante à primeira matriz e imprimir o resultado

Cada item deve ser implementado em uma função diferente e as opções (1) a (4) devem verificar se as matrizes já estão criadas antes de serem executadas.

8. Faça um programa que leia/crie duas matrizes A e B de tamanho 3 x 3 e

- calcule $C = A * B$
- calcule $C = A^2$
- calcule $C = A^n$ (n fornecido pelo usuário)

Modifique o programa para que funcione para matrizes A e B de tamanho $m \times m$

9. Implemente uma função `main()` que leia ou inicialize um vetor de 10 posições contendo o gabarito de respostas que podem ser a, b, c ou d. Crie aleatoriamente uma matriz 5 x 10 que se refere às respostas de 10 questões de múltipla escolha, fornecidas por 5 alunos (vc pode utilizar a função `criaMatriz(...)` adaptada para converter os números em letras). Crie uma função para comparar as respostas de cada candidato com o gabarito e retornar (por referência) um vetor denominado resultado, contendo a pontuação correspondente a cada aluno.

10. Refaça os exercícios de 1 a 5 agora utilizando funções para criar as matrizes, imprimir as matrizes, realizar as operações solicitadas em cada exercício.